REMARKS

Applicants respectfully request the Examiner's reconsideration of the present application as currently amended.

Claims 1, 3, and 5-24 in the present application are pending.

Claims 1, 3, 5-21, and 23-24 are rejected under 35 U.S.C. §112, first paragraph.

Claims 1, 3, and 5-21 are rejected under 35 U.S.C. §112, second paragraph.

Claims 1, 3, 5-7, 9, and 11-24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,711,602 ("Bhandal") in view of U.S. Patent No. 7,046,723 ("Schier").

Claims 8 and 10 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claims 1, 3, 5-7, 9, and 11-22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over 6,711,602 ("Bhandal") in view of U.S. Patent 7,046,723 ("Schier").

Claims 1, 11, 17, 21, and 22 have been amended.

Claims 25 and 26 have been added.

Support for amended claims 1, 11, 17, 21, and 22 and new claims 25 and 26 may be found at paragraphs [0018]-[0019], [0028], [0031], and [0037]-[0038] in the specification, and Figures 1-4 of the drawings. No new matter has been added.

Claims 1, 3, 5-21, and 23-24 are rejected under 35 U.S.C. §112, first paragraph, as failing to comply with the written description requirement.

Claims 1, 3, and 5-21 are rejected under 35 U.S.C. §112, second paragraph as being indefinite for failing to particularly point out and distinctly claim he subject matter which applicant regards as the invention.

The Office specifically states the following.

> Re claim 1, the limitation "DSP ONLY capable of supporting ... number of bits that are fewer in number than..." is not fully support or described ...
>
> ...
>
> Re claim 1, the term "capable of" in line 7 is a relative term which renders the claim indefinite.

(414/2010 Office Action, pp. 2-3).

Claims 1, 11, 17, and 21 have been amended. The "ONLY " and "capable of" language in the claims have been deleted.

Applicants submit that in view of the amendments to claims 1, 11, 17, and 21 that claims 1, 3, 5-21, and 23-24 comply with 35 U.S.C. §112, first and second paragraphs.

It is submitted that Bhandal and Schier do not render claims 1, 3, 5-7, 9, and 11-24 unpatentable under 35 U.S.C. §103(a).

Bhandal includes a disclosure of a pair of parallel 16.times.16 multipliers each with two 32-bit inputs and one 32-bit output. There are options to allow input halfword and byte selection for four independent 8x8 or two independent 16x16 multiplications, real and imaginary parts of complex multiplication, pairs of partial sums for 32x32 multiplication, and partial sums for 16x32 multiplication. There are options to allow internal hardwired routing of each multiplier unit results to achieve partial-sum shifting as required to support above options. There is a redundant digit arithmetic adder before final outputs to support additions for partial sum accumulation, complex multiplication vector accumulation and general accumulation for FIRs/IIRs--giving MAC unit functionality. There are options controlled using bit fields in a control register passed to the multiplier unit as an operand. There are also options to generate all of the products needed for complex multiplication (see Bhandal Abstract).

Schier includes a disclosure of a digital and a multiplication method are described, which lead to an efficient architecture for a hardware implementation of digital FIR and IIR filters into FPGAs. The multiplications of input sample data and delayed sample data with filter

coefficients are performed by addressing look-up tables in which corresponding multiplication results are prestored. The size of the look-up tables is reduced by storing only those multiplication results which cannot be obtained by a shifting operation performed on the other pre-stored multiplication results, the input sample data, or the delayed sample data. Thereby, the size of the look-up tables can be compressed significantly such that an implementation of large digital filters into FPGAs is possible (see Schier Abstract).

It is submitted that Bhandal and Schier do not teach or suggest

> A method for performing multiplication of a first number with a second number on a target device, comprising:
> generating a product by multiplying a first plurality of bits from the first number and a first plurality of bits from the second number using a single digital signal processor (DSP) where a largest dimension of multiplication supported by the DSP is under that which supports multiplying the first and second numbers;
> retrieving a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory where the second plurality of bits from the first number is fewer than the bits of the first number and the second plurality of bits from the second number is fewer than the bits of the second number;
> scaling the product with respect to a position of the first plurality of bits from the first number and a position of the first plurality of bits from the second number and scaling the stored value with respect to a position of the second plurality of bits from the first number and a position of the second plurality of bits from the second number; and
> summing a scaled product and a scaled stored value to generate a value representing a product of the first number and the second number, wherein the first number and the second number each have a number of bits equal to or greater than a total of the first and second plurality of bits.

(Claim 1, as Amended) (Emphasis Added).

On the contrary, Applicants submit that Bhandal discloses a method of multiplying a first number with a second number using a digital signal processor (DSP) 44 where the dimension of multiplication supported by the DSP supports multiplying the first and second number. Bhandal discloses a DSP 44 (shown in Figures 1 and 2) that includes a plurality of M-unit groups 84 each

having a M Galois multiply unit 164 and a M multiply unit 171 (shown in Figure 5). Figure 11A-I illustrate various multiply operations performed by a pair of parallel 16x16 multipliers to allow for four independent 8x8 multiplications, two independent 16x16 multiplications, 32x32 multiplication, and 16x32 multiplication (see Bhandal column 5, lines 11-15, column 6, lines 15-27, column 8, lines 3-10, column 8, line 40 through column 9, line 4, and Abstract, and Figures A-I). For all of the multiplication operations described and illustrated in Bhandal, the DSP 44 supports a dimension of multiplication that allows for the multiplication of the first and second number.

In the Office Action mailed 4/14/2010, the Office states in part the following.

> The examiner respectfully submits that the above rejection fully addresses all the limitations in the claims, particularly the newly added limitations wherein the number of bits of high or low portion of the source is less than the whole bits of source as seen in Figures 11-12 wherein either high or low portion is only 16 bits compare to 32 bits of the source. Generally, Figures 11 show multiple way of performing multiplications wherein the multipliers and multiplicands are less then (sic) the sources or the first and second numbers as cited in the claimed invention. For instant, Figure 11B show first numbers as 32-bit source 1 and second number as 32-bit source 2 and the multiplication is performed on either lower or upper portion of the first and second numbers to produce AC and/or BD. Thus, clearly DSP is capable of supporting multiplication on a number of bits that are fewer in number than that forming the first and second n umbers as seen in Figures 11.

(4/14/2010 Office Action, p. 16) (Emphasis Added).

Applicants respectfully disagree. Applicants submit that the 16x16 multiplication operation that is broken down to (A*C) + (B*D) in Figure 11B is performed by a single DSP 44 in Bhandal which supports the dimension of multiplication (16x16) that allows for the multiplication of the first and second numbers (see Bhandal column 5, lines 11-15, column 6, lines 15-27, column 8, lines 3-10, column 8, line 40 through column 9, line 4, and Abstract, and Figure 11B).

Schier only discloses a digital filter and method for performing a multiplication based on a look-up table.

Furthermore, the Office acknowledges that Bhandal does not disclose retrieving a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory where the second plurality of bits from the first number is fewer than the bits of the first number and the second plurality of bits from the second number is fewer than the bits of the second number (see 4/14/2010 Office Action, p. 5). Schier discloses a LUT based multiplier where the entire result of multiplication of sample data with filter coefficients are prestored. Input sample data x(n) is not split up into their bit positions, but are completely supplied to the LUT based multiplier 2 in order to address a look-up table corresponding to the respective filter coefficient and pre-storing result of multiplications (see Schier column 5, lines 29-35, and Figure 1). Schier does not prestore only a portion of a result. Weighted sample data generated from multiplying sample data with filter coefficients are outputted from the LUT based multiplier 2 (see Schier column 7, line 65 through column 8, line 29 and Figures 1 and 4).

Moreover, Applicants submit that there is no basis for combining Bhandal and Schier. In the Office Action mailed 4/14/2010, the Office states in part the following.

> Bhandal et al. fail to disclose the multiplier is on a field programmable gate array and that the second product is retrieved from a memory. However, Schier et al. disclose in Figures 1-4 the multiplier is on a field programmable gate array (e.g. abstract) and the second product is retrieved from a memory (e.g. any intermediate product from the LUT in Figures 1-4 as blx).
>
> Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention is made to add the multiplier is (sic) on a field programmable gate array and the second product is retrieved from a memory as seen in Schier et al.'s invention into Bhandal et al's invention because it would enable to improve system performance (e.g. col. 3 lines 9-11 and col. 4 lines 9-12)

(4/14/2010 Office Action, p. 5) (Emphasis Added).

Applicants submit that the Office cannot properly combine "any intermediate product from the LUT in Figures 1-4" of Schier with Bhandal to be used as "the second product" as the Office suggests on page 5 of the Office Action mailed 4/14/2010. In Schier, the values b0x(n) to b4x(n-4) are not "intermediate products". The values are final products that do not require further processing to generate a multiplication product. Adder 3 sums the multiplication products b0x(n) to b4x(n-4) to generate filter output data y(n), not to generate a multiplication result. By retrieving from memory "any intermediate product from the LUT in Figures 1-4" of Schier as the Office suggests would render the invention in Schier unsatisfactory for its intended purpose (See MPEP 2143.01 V)

Applicants also disagree with the Office's motivation for combining Bhandal with Schier when the Office states that "it would have been obvious to a person having ordinary skill in the art at the time of the invention is made to add ... the second product is (sic) retrieved from a memory as seen in Schier et al.'s invention into Bhandal et al.'s invention because it would enable to improve system performance." Applicants submit that the example in Bhandal discloses that the number of bits of the numbers being multiplied, 32, match the configuration of the number of bits multiplied by multipliers 800 and 801, 32. Thus, there is no need or motivation to add a further product from memory. Furthermore, because there is a match, there would be no improvement in system performance. In fact, by requiring a stored product to be added, the multiplier in Bhandal would have its performance worsen not improve.

In the Office Action mailed 4/14/2010, the Office states in part the following.

> The examiner respectfully submits that the rejection clearly states the motivation for combining the references and obviously one ordinary skilled in the art would known (sic) the benefits of having multiplication by retrieving product from table/memory instead of actual performing direct multiplication of product from table/memory instead of actual performing direct multiplication of product. One quick lookup at Google produces this link for obvious advantage of having LUT instead direct multiplication at http://www.coranac.com/tonc/text/fixed.htm.

(4/14/2010 Office Action, p. 18) (Emphasis Added).

Applicants respectfully submit that the Office's interpretation of the document at the link provided by the Office in fact teaches away from the combination of Bhandal and Schier. According to the Office, there is an advantage of "having LUT instead direct multiplication". Using the Office's logic, since Schier discloses a more advantageous method of performing multiplication by using look-up tables, there would then be no motivation for using any partial results from the DSP disclosed in Bhandal.

Applicants further submit that the document at the link provided by the Office is dated February 8, 2007, after the filing date of the present application. The cited document would not have been available to one skilled in the art at the time the present application was filed.

In contrast, claim 1 states

> A method for performing multiplication of a first number with a second number on a target device, comprising:
> generating a product by multiplying a first plurality of bits from the first number and a first plurality of bits from the second number using a single digital signal processor (DSP) where a largest dimension of multiplication supported by the DSP is under that which supports multiplying the first and second numbers;
> retrieving a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory on the target device where the second plurality of bits from the first number is fewer than the bits of the first number and the second plurality of bits from the second number is fewer than the bits of the second number;
> scaling the product with respect to a position of the first plurality of bits from the first number and a position of the first plurality of bits from the second number and scaling the stored value with respect to a position of the second plurality of bits from the first number and a position of the second plurality of bits from the second number; and
> summing a scaled product and a scaled stored value to generate a value representing a product of the first number and the second number, wherein the first number and the second number each have a number of bits equal to or greater than a total of the first and second plurality of bits.

(Claim 1, as Amended) (Emphasis Added).

Claims 21 and 22 include similar limitations. Claims 11 and 17 include limitations similar to retrieving/storing a stored value designated as a product of a second plurality of bits from the first number and a second plurality of bits from the second number from a memory where the second plurality of bits from the first number is fewer than the bits of the first number and the second plurality of bits from the second number is fewer than the bits of the second number.

Given that claims 3, 5-10, 12-16, 18-20, and 23-24 depend from claims 1, 11, and 17, , it is likewise submitted that claims 3, 5-10, 12-16, 18-20, and 23-26 are also patentable under 35 U.S.C. §103(a) over Bhandal and Schier.

Applicants submit that Bhandal and Schier also do not teach or suggest

> The method of Claim 1, wherein scaling the product comprises routing the product directly to an adder at inputs of appropriate significance.

(Claim 23) (Emphasis Added).

In the Office Action mailed 4/14/2010, the Office states in part the following.

> The examiner respectfully submits that scaling and routing to appropriate significance or position for summing must be exits in multiplication, otherwise it would produce wrong result of multiplications. Figures 11 show multiple configuration of multiplications wherein the result of certain products are shifted as scaled or routed to correct significance for summing.

(4/14/2010 Office Action, p. 18) (Emphasis Added).

Applicants respectfully submit that Figures 11A-I show "optional scaling", but does not disclose how the "optional scaling" is implemented. Applicants direct the Office to Figure 8 of Bhandal. In the Office Action mailed 6/10/2009, the Office acknowledges that Figure 8 of Bhandal discloses scaling of a product being achieved "by shifter 810 in Figure 8" (4/14/2010 Office Action, p. 14). Bhandal illustrates routing an output of multiplier 800 to a shifter 810 before routing the output to adder/converter 820 (see Bhandal column 7, lines 31-50 and Figure

8). Clearly, the product from multiplier 800 is not scaled by being routed directly to adder/converter 820 at inputs of appropriate significance.

Schier only discloses a method for performing a multiplication based on a look-up table. Schier does not teach or suggest scaling a product with respect to a position of the first plurality of bits from the first number and a position of the first plurality of bits from the second number wherein scaling the product comprises routing the product directly to an adder at inputs of appropriate significance.

In contrast, claim 23 states

> The method of Claim 1, wherein scaling the product comprises routing the product directly to an adder at inputs of appropriate significance.

(Claim 23, Emphasis Added).

Claims 11 and 21 include similar limitations.

Applicants submit that Bhandal and Schier also do not teach or suggest

> The method of Claim 1, wherein scaling the stored value comprises routing the stored value directly to an adder at inputs of appropriate significance.

(Claim 24, Emphasis Added).

On the contrary, Schier discloses a LUT 2 that routes output data stored from a reduced table 210/215 directly to a bit shifting units 220/225 where a bit shifting operation is performed (see Schier column 8, lines 18-23 and Figure 4). The out data is transmitted to the bit shifting units 220/225 before being output to adder 3.

As stated above, the Office acknowledges that Figure 8 of Bhandal discloses scaling of a product being achieved "by shifter 810 in Figure 8" (4/14/2010 Office Action, p. 14). Bhandal does not teach or suggest scaling a stored value by routing the stored value directly to an adder at inputs of appropriate significance.

In contrast, claim 24 states

The method of Claim 1, wherein scaling the stored value comprises routing the stored value directly to an adder at inputs of appropriate significance.

(Claim 24, Emphasis Added).

Claims 11 and 21 include similar limitations.

In view of the arguments set forth herein, it is respectfully submitted that the applicable rejections and have been overcome. Accordingly, it is respectfully submitted that claims 1, 3, and 5-24 should be found to be in condition for allowance.

The Examiner is invited to telephone Applicants' attorney (217-377-2500) to facilitate prosecution of this application.

If any additional fee is required, please charge Deposit Account No. 50-1624.

Respectfully submitted,

Dated: August 14, 2010

Lawrence M. Cho
Attorney for Applicants
Registration No. 39,942

P.O. Box 2144
Champaign, IL 61825
(217) 377-2500